Please **AMEND** the claims as follows:

1. (Currently Amended) In a compiler, a method of generating assembly code to aid in stack unwinding of a memory stack, the method comprising:

obtaining ~~one or more~~ at least one source code line ~~lines~~; and

generating assembly code for the ~~one or more~~ at least one source code line ~~lines~~, the assembly code including ~~one or more~~ at least one stack unwind assembler directive ~~directives~~, each of the stack unwind assembler directives having ~~one or more~~ at least one associated stack unwind sub directive ~~directives~~, each ~~of the~~ stack unwind assembler directive ~~directives~~ being adapted for indicating to an assembler that ~~one or more~~ at least one encoded data section ~~sections~~ containing information to be used during stack unwinding is to be generated in an object file from the ~~one or more~~ at least one associated stack unwind sub directive ~~directives.~~, thereby enabling the assembler to generate the at least one encoded data section to be used during stack unwinding, wherein each stack unwind assembler directive is a human readable indicator indicating to the assembler that associated sub directives are related to stack unwinding, wherein the stack unwind assembler directive does not specify a function to be performed during stack unwinding.

2. (Original) The method as recited in claim 1, wherein each stack unwind sub directive indicates a stack unwind operation to be performed.

3. (Original) The method as recited in claim 1, wherein each stack unwind sub directive indicates a previously performed stack operation for which unwinding is to be performed.

4. (Currently Amended) The method as recited in claim 1, wherein each stack unwind sub directive indicates ~~one or more~~ at least one stack operation ~~operations~~ that are to be reversed by a stack unwind mechanism using the ~~one or more~~ at least one encoded data section ~~sections~~.

5. (Original) Assembly code generated according to the method of claim 1.

6. (Currently Amended) In a compiler, a method of generating assembly code to aid stack unwinding of a memory stack, the method comprising:

obtaining ~~one or more~~ at least one source code line ~~lines~~; and

generating from the ~~one or more~~ at least one source code line ~~lines~~ a stack unwind assembler directive and an associated stack unwind sub directive, wherein the stack unwind assembler directive indicates that ~~one or more~~ at least one encoded data section ~~sections~~ containing stack information to be used for stack unwinding of the stack is to be generated by an assembler from the stack unwind sub directive, wherein each stack unwind assembler directive is a human readable indicator indicating to the assembler that associated sub directives are related to stack unwinding, wherein the stack unwind assembler directive does not specify a function to be performed during stack unwinding.

7. (Currently Amended) In a compiler, a method of generating assembly code to aid stack unwinding from a set of source code, the method comprising:

generating a stack unwind assembler directive; and

generating a stack unwind sub directive, wherein the stack unwind assembler directive indicates that ~~one or more~~ at least one encoded data section ~~sections~~ containing stack information to be used for unwinding of a stack is to be generated by an assembler from the stack unwind sub directive, wherein each stack unwind assembler directive is a human readable indicator indicating to the assembler that associated sub directives are related to stack unwinding, wherein the stack unwind assembler directive does not specify a function to be performed during stack unwinding.

8. (Original) The method as recited in claim 7, wherein the stack unwind sub directive indicates a region operation designating one or more portions of a function.

9. (Original) The method as recited in claim 8, wherein the region operation designates a prologue region of a function or a body region of a function.

10. (Original) The method as recited in claim 7, wherein the stack unwind sub directive indicates which registers have been saved prior to a function call.

11. (Original) The method as recited in claim 7, wherein the stack unwind sub directive indicates whether the stack has a fixed size or a variable size.

12. (Original) The method as recited in claim 7, wherein the stack unwind sub directive indicates a number of entries in the stack.

13. (Original) The method as recited in claim 7, wherein the stack unwind sub directive indicates whether a function being called has a fixed number of arguments or a variable number of arguments.

14. (Original) The method as recited in claim 11, wherein when the stack has a variable size, the stack unwind sub directive further indicates that a stack pointer is saved.

15. (Original) The method as recited in claim 11, wherein when the stack has a variable size, the stack unwind sub directive further indicates that a stack pointer is restored.

16. (Original) The method as recited in claim 7, wherein the stack unwind sub directive indicates a source register from which data is to be obtained.

17. (Original) The method as recited in claim 16, wherein the stack unwind sub directive further indicates a destination register or destination memory location to which the obtained data is to be saved.

18. (Original) The method as recited in claim 17, wherein the destination memory location is indicated through an offset from a stack pointer or previous stack pointer.

19. (Original) The method as recited in claim 7, wherein the stack unwind sub directive indicates a memory location or first register to which contents of a second register is to be spilled.

20. (Original) The method as recited in claim 19, wherein the memory location is specified relative to a stack pointer or previous stack pointer.

21. (Original) The method as recited in claim 19, wherein the second register is of type branch, floating point, or general purpose.

22. (Original) The method as recited in claim 19, wherein the stack unwind sub directive indicates multiple destination registers to which contents of one or more source registers are to be spilled.

23. (Currently Amended) A computer-readable medium for generating assembly code for stack unwinding of a memory stack, comprising:

instructions for obtaining ~~one or more~~ at least one source code line ~~lines~~; and

instructions for generating from the ~~one or more~~ at least one source code line ~~lines~~ a stack unwind assembler directive and an associated stack unwind sub directive, wherein the stack unwind assembler directive indicates that ~~one or more~~ at least one encoded data section ~~sections~~ containing information to be accessed during stack unwinding of the stack is to be generated by an assembler from the stack unwind sub directive, wherein each stack unwind assembler directive is a human readable indicator indicating to the assembler that associated sub directives are related to stack unwinding, wherein the stack unwind assembler directive does not specify a function to be performed during stack unwinding.

24. (Currently Amended) An apparatus for generating assembly code for stack unwinding of a memory stack, comprising:

a processor; and

a memory, at least one of the processor and the memory being adapted for:

obtaining ~~one or more~~ at least one source code line ~~lines~~; and

generating from the ~~one or more~~ at least one source code line ~~lines~~ a stack unwind assembler directive and an associated stack unwind sub directive, wherein the stack unwind assembler directive indicates that ~~one or more~~ at least one encoded data section ~~sections~~ containing stack information to be used to perform stack unwinding of the stack is to be generated by an assembler from the stack unwind sub directive, wherein each stack unwind assembler directive is a human readable indicator indicating to the assembler that associated sub directives are related to stack unwinding, wherein the stack unwind assembler directive does not specify a function to be performed during stack unwinding.

25. (Currently Amended) An apparatus for generating assembly code for stack unwinding of a memory stack, comprising:

means for obtaining ~~one or more~~ at least one source code line ~~lines~~; and

means for generating from the ~~one or more~~ at least one source code line ~~lines~~ a stack unwind assembler directive and an associated stack unwind sub directive, wherein the stack unwind assembler directive indicates that ~~one or more~~ at least one encoded data section ~~sections~~ containing information to be used for stack unwinding of the stack is to be generated by an assembler from the stack unwind sub directive, wherein each stack unwind assembler directive is a human readable indicator indicating to the assembler that associated sub directives are related to stack unwinding, wherein the stack unwind assembler directive does not specify a function to be performed during stack unwinding.

26. (Currently Amended) In an assembler, a method of generating object code from assembler stack unwind directives, the method comprising:

receiving a stack unwind assembler directive having ~~one or more~~ at least one associated stack unwind sub directive ~~directives~~, wherein the stack unwind assembler directive indicates to the assembler that ~~one or more~~ at least one encoded data section ~~sections~~ containing information for implementing stack unwinding is to be generated from the ~~one or more~~ at least one associated stack unwind sub directive ~~directives~~; and

generating ~~one or more~~ at least one encoded data section ~~sections~~ containing stack information to implement a stack unwind feature designated by the ~~one or more~~ at least one stack unwind sub directive ~~directives~~, wherein each stack unwind assembler directive is a human readable indicator indicating to the assembler that associated sub directives are related to stack unwinding, wherein the stack unwind assembler directive does not specify a function to be performed during stack unwinding.

27. (Currently Amended) The method as recited in claim 26, wherein generating ~~one or more~~ at least one encoded data section ~~sections~~ containing information to implement the stack unwind feature designated by the ~~one or more~~ at least one stack unwind sub directive ~~directives~~ comprises:

generating ~~one or more~~ at least one encoded data section ~~sections~~ containing information to reverse stack operations previously performed as indicated by the ~~one or more~~ at least one stack unwind sub directive ~~directives~~.

PLEASE ADD NEW CLAIMS AS FOLLOWS:

28. (New) The method of claim 1, wherein each stack unwind assembler directive is the same stack unwind assembler directive.

29. (New) The method as recited in claim 1, wherein each stack unwind assembler directive is only an indicator indicating to the assembler that associated sub directives are related to stack unwinding.

30. (New) The method as recited in claim 1, wherein each stack unwind assembler directive does not direct the assembler to perform a function related to stack unwinding.

31. (New) The method as recited in claim 1, wherein each sub directive directs the assembler to perform a function related to stack unwinding.

32. (New) The method as recited in claim 1, wherein each stack unwind assembler directive is not a jump instruction.

REMARKS

In the Office Action, the Examiner objected to the drawings, and rejected the claims under 35 USC §112, 35 USC §102 and 35 USC §103. In response, the drawings have been amended to correct the typographical errors noted. It appears that the Examiner has objected to the use of the phrase "one or more" in the rejection of the claims under 35 USC §112. As a result, Applicant has amended the phrase "one or more" to "at least one." Therefore, the meaning of the claims remains the same. In addition, Applicant has amended the claims to further clarify the subject matter regarded as the invention. Moreover, claims 28-32 have been added. Claims 1-32 are now pending. Reconsideration of the application is respectfully requested based on the following remarks.